Scalable Computing: Practice and Experience, ISSN 1895-1767, http://www.scpe.org
© 2025 SCPE. Volume 26, Issues 2, pp. 531–539, DOI 10.12694/scpe.v26i2.3946

IMPROVEMENT OF THE ACO ALGORITHM FOR INTELLIGENT TASK SCHEDULING IN CLOUD SYSTEMS

ESMA INSAF DJEBBAR* AND GHALEM BELALEM†

Abstract. Cloud computing involves accessing and using computing resources, such as servers, storage, and software applications, over the Internet, enabling scalable access on demand. Cloud computing systems are becoming an essential platform for scientific applications. They enable task scheduling and IT resource allocation. When these resources are not enough to meet demand, planning techniques are necessary. We propose to apply an improved ACO algorithm for intelligent task scheduling and appropriate resource allocation in a cloud environment. This work proposes a modified version of the ACO algorithm that can quickly converge to the best solution to further optimize the total response time, the average response time and the total execution cost. The algorithm suggested based on artificial intelligence application of enhanced ACO is compared with the classical ACO algorithm using Cloudsim simulator. The results obtained after the experiments and simulation are very encouraging to adopt this technique.

Key words: Intelligent Task Scheduling, improved ACO Algorithm, Cloud computing Systems, Cloudsim simulator.

1. Introduction. Cloud computing is an area that brings together the distribution technologies, on demand and via the Internet, of computer software and hardware services. The central idea behind these techniques is to distribute computing resources as a common necessity, as envisioned by the founders of the revolutionary technology over 40 years ago [1]. This principle of public distribution of computing resources also drives the grid computing community, so it is sometimes difficult to distinguish the boundary between Grid and Cloud Computing. This difficulty is all the more real since cloud computing is a young concept, the first implementations of which date back to 2006, and whose development has accelerated in recent years.

Cloud computing reflects a recent model for the provision of computing resources. This model facilitates access to resources via the network in sequence to minimize the costs incurred in managing hardware and software resources. It constitutes the importance of viewing computing as a service in which principled reduction can significantly minimize the cost of computing resources. With the cloud, there is no point investing in an Infrastructure that would be overpriced to acquire. Since the contribution is menstrual, you have better surveillance of your budget and you only pay for what you use. You no longer have to worry about updates, storage and performance issues. Using the cloud, all this is managed by your service provider. The applications and services you use in the Cloud are accessible wherever you are as long as you have a terminal and an internet connection. If your needs change, it is possible to adapt your offer quickly and simply. The user can personalize the services present in his interface.

The concept of Scheduling tasks within cloud computing systems is garnering more attention as Cloud technology becomes increasingly popular. Task scheduling typically involves organizing tasks and assigning them to available resources based on their properties and requirements. This process is vital for ensuring the efficient operation of Cloud systems, as it involves considering various task modules to schedule tasks appropriately. It's essential to utilize the appropriate resources cost-effectively without changing the settings of the Cloud service.

Various algorithms can be employed to select the necessary resources in a cloud system. Resource sorting methods may include random selection, Round Robin, or greedy algorithms (based on resource processing capacity and waiting time), among others. Similarly, task sorting can be based on FCFS (First Come, First

*Department of Computer Science Systems Engineering, National Polytechnic School of Oran-Maurie Audin, Oran, Algeria (esma.djebbar@gmail.com).
†Department of Computer Science, University of Oran1-Ahmed Ben Bella, Algeria

Served), SJF (Shortest Job First), or priority-based approaches. The scheduling algorithm determines which task to execute and on which available resource. Each selection strategy offers distinct advantages, and by implementing them strategically, it's possible to harness their benefits and mitigate the drawbacks, ultimately aiming for an optimized solution.

The remainder of the article is structured as follows: Section 2 introduces the Ant Colony Optimization (ACO) algorithm. Section 3 reviews related studies on task scheduling utilizing the ACO algorithm in cloud computing. Section 4 details the proposed Intelligent Ant Colony Optimization Task Scheduling algorithm. Section 5 presents the experimental results conducted in the Cloudsim simulator. Lastly, Section 6 wraps up the article by summarizing our contributions and suggesting potential avenues for future research.

2. Fundamentals of Ant Colony Algorithm Programming. Ant colony algorithms are derived from the behavior of ants and constitute a collection of optimization meta-heuristics.
The ant colony optimization algorithm operates as an iterative process involving individuals that communicate similar findings, enabling them to guide their subsequent decisions and signal paths to follow or discard to subsequent individuals.

This concept originates from observing how ants search for food resources. Despite their limited individual cognitive abilities, ants collectively manage to discover the most efficient route between a food source and their nest.

The ACO organization chart is provided as a follow-up (Figure2.1). A framework describing this operation is:

1. An ant circulates almost randomly around its colony space.
2. If it detects a feeding orovenance, it usually returns immediately to the nest, releasing a trail of pheromones in its wake.
3. These pheromones being endearing, ants passing through the surrounding area will have the instinct to pursue the imprint, more or less directly

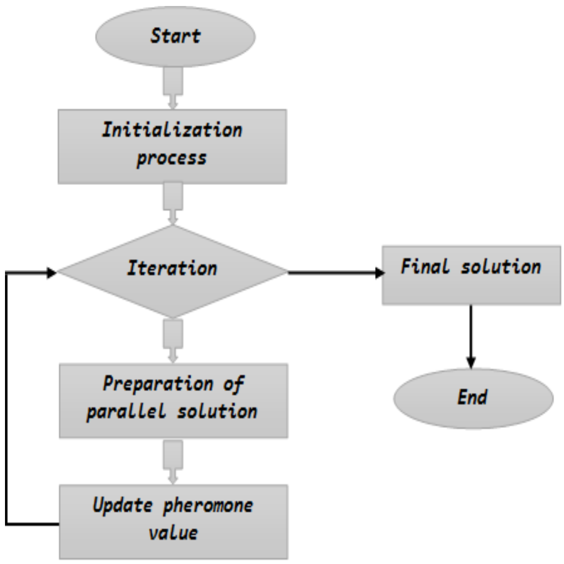

Fig. 2.1: Model of basic ACO Algorithm

3. Related works. Ant colony algorithms (ACO) for optimization can give approximate solutions to difficult problems. In this part, we summarize some works that have used the ACO algorithm for scheduling in cloud systems. Researchers have provided satisfactory algorithms and methods for task scheduling and resource allocation, but there is still room for imperfection since the methods and algorithms mentioned are complex

Table 3.1: Comparison of Task schedulinng algorithm with proposed work.

Work	Objective	Used algorithm	Key contribution	Results
[10].	Ant Colony Optimization for Scientific Workflow.	A Genetic Algorithm and Ant Colony Optimization Algorithm.	Implement Ant Colony Optimization (ACO), initializing the allocation of underutilized VMs with the Pareto distribution.	Reduce response time.
[11].	Optimizing Multi Objective Based Dynamic Workflow.	Black Hole Algorithm and Ant Colony Optimization algorithm.	It gives the hybrid approach for proposed algorithm.	Reduce makespan and total execution cost.
[12].	Enhancing the Ant Colony Optimization Algorithm for Improved Performance.	A* Algorithm and Ant Colony Optimization algorithm.	Introducing the ACOStar algorithm aims to augment ACO performance by integrating the evaluation function used in the A* algorithm.	Reduce execution time.
[13].	An optimized resource scheduling algorithm in fog computing.	A Genetic Algorithm and Ant Colony Optimization algorithm.	Propose New Genetic Ant Colony Optimization Algorithm (NGACO).	Reduce makespan, economic cost, and total cost.
[14].	A hybrid algorithm for deadline-constrained workflow scheduling.	A Genetic Algorithm and Ant Colony Optimization Algorithm.	Introduce a hybrid scheduling algorithm, Partial Critical Path–Ant Colony Optimization (PCP–ACO).	Reduce execution cost.
[15].	Ant colony optimization for parallel test assembly.	Ant Colony Optimization algorithm.	Showcase the versatility of the ACO to construct multiple parallel short scales.	Reduce parallel test assembly.
[16].	An enhanced algorithm for service composition based on ACO.	Ant Colony Optimization algorithm.	An Ant Colony Optimization (ACO) algorithm that incorporates a multi-pheromone approach.	Improve the quality of service (QoS).
[17].	An adaptive ACO algorithm with dynamic ant quantity for addressing the traveling salesman problem.	Dynamic adaptive ant quantity ACO Algorithm and Ant Colony Optimization algorithm.	Dynamically determine the number of ants to prevent them from falling into local optimization.	Reduce convergence time and increase solution quality.
Proposed.	Improvement of the ACO Algorithm for Intelligent Task Scheduling.	Ant Colony Optimization algorithm and Improved ACO Algorithm.	Select the best virtual machine for running tasks.	Reduce response time and execution cost.

class problems (NP-difficult/NP-complete). Table 3.1 provides a comparison of the studied works on task scheduling in cloud computing systems presenting their objectives, algorithms, key contributions and specific results.

The authors in [10] use the Ant Colony Algorithm (ACO) intelligent optimization, in which the allocation of underutilized virtual machines is initiated by the Pareto law repartition. The ACO program is chosen to lead to the determination of migration of virtual machines (VM) by its outcome towards minimum values of cost and response time.

In their study [11], the authors propose enhancing dynamic task scheduling in the Cloud by employing two distinct scheduling algorithms: the Ant Colony Optimization and the Black Hole Algorithm. Their objective is to reduce overall costs and total response time for users. They achieve this by integrating the Black Hole Algorithm with the ACO program, replacing certain steps.

In their research [12], the authors introduce the ACOStar algorithm, aiming to enhance the efficiency of ACO by incorporating an estimation function into its mathematical framework. To validate the performance of this algorithm, they apply it to the shortest path problem, demonstrating its effectiveness and reliability through practical tests.

In [13], the authors propose a resource scheduling model called Normalization Processing, which optimizes pheromone levels to minimize total costs. They also introduce the New Genetic Ant Colony Optimization (NGACO) Algorithm, combining enhancements from Genetic Algorithm and ACO. NGACO features a novel pheromone update approach and utilizes the roulette wheel algorithm for enhanced exploration and population diversity.

The hybrid scheduling algorithm, Partial Critical Path-Ant Colony Optimization (PCP-ACO), is introduced in [14] with the aim of reducing workflow execution costs in cloud environments while meeting user-defined deadlines. By merging the heuristic PCP algorithm with ACO, PCP-ACO achieves faster convergence.

In [15], the authors demonstrate ACO's adaptability in constructing multiple parallel short scales that fulfill various criteria simultaneously. They curate tests from a pool of knowledge items, ensuring gender fairness and precise measurement of factual knowledge.

In [16], a service composition strategy in a multi-cloud environment integrates an Ant Colony Optimization (ACO) algorithm with a multi-pheromone mechanism to enhance Quality of Service (QoS). Additionally, a mutation operation inspired by genetic algorithms is incorporated to mitigate the risk of local optima.

In [17], the authors present the dynamic adaptive ACO algorithm (DAACO), which utilizes a hybrid local selection strategy to enhance the quality of ant optimization and reduce optimization time. Experimental results on the Traveling Salesman Problem (TSPLIB dataset) demonstrate that DAACO outperforms existing ACO algorithms in terms of convergence time, solution quality, and average performance.

4. Proposition of modified Algorithm Ant Colony Optimization Task Scheduling. We propose applying an enhanced ACO algorithm for intelligent task scheduling and optimal resource allocation in a cloud environment. This study introduces a modified version of the ACO algorithm aimed at further optimizing total response time, average time, and overall execution costs. The proposed algorithm, leveraging enhanced ACO through artificial intelligence techniques, is compared against the traditional ACO algorithm using the Cloudsim simulator. A graphical representation of our approach is illustrated in Figure 4.1.

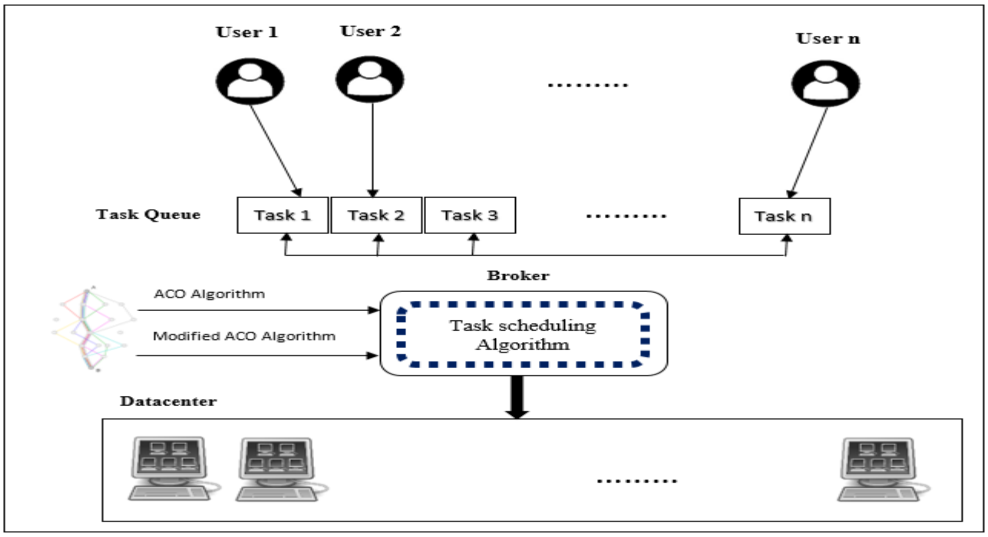

Fig. 4.1: The graphical representation of the proposed work

The improved ACO algorithm for selecting the best virtual machines for task execution and which allows rapid convergence towards the best solution and which is composed of three phases:

4.1. The choice of the VM for the task. For each repetition of the ACO algorithm, and for each individual ant k, $k = 1, ..., m$ (m is the quantity of ants), creates a round carrying out n steps (n is the set of tasks) of which a probability equation is used. The k-th ant selects Vm_j for the following task i with a probability calculated by equation 4.1:

$$P_{ij}^{k}(t) = \begin{cases} \frac{[\tau_{ij}(t)]\times\alpha^2\times[\eta_{ij}(t)]\times\beta^2}{\sum_{j\in authorized_k}[\tau_{ij}(t)]\times\alpha^2\times[\eta_{ij}(t)]\times\beta^2} & \text{if } j \in authorized_k \\ 0 & \text{else.} \end{cases} \tag{4.1}$$

$\tau_{ij}(t)$ indicates the concentration of pheromones at time t between task i and vm_j on the path.
η_{ij} represents visibility; the expected execution and transmission times of task i on the virtual machine vm_j.It is possibly estimated using the following equation 4.2:

$$\eta_{ij} = \ 1/C_{ij} \tag{4.2}$$

C_{ij} is the completion time. $authorized_k$ represents the virtual machines authorized for the ant k. $k = \{0, 1, ..., n-1\}$.
α and β are two parameters that look at weight for pheromone trail and visibility information respectively.

4.2. Pheromone Update. When a turn is made, equation 4.3 calculates the quantity of pheromones on each path (i, j), deposited by each ant k:

$$\Delta\tau_{ij}^{k}(t) = \begin{cases} \frac{Q}{L} & \text{if}(i,j) \in T_k(t) \\ 0 & \text{else.} \end{cases} \tag{4.3}$$

$T_k(t)$ represents for each iteration t the turn performed by the ant k. L_k calculated by equation 4.4, defines the length of the turn:

$$L_k = \ argmax_j \left\{\sum_{i\in IJ} C_{ij}\right\} \tag{4.4}$$

IJ is the totality of tasks assigned to virtual machine vm_j. Q is a adjustment coefficient.

The pheromone update employed at all paths is updated by equation 4.5 after each iteration:

$$\tau_{ij}(t+1) = \ (1-\rho)\tau_{ij}(t) + \rho\Delta\tau_{ij}(t) \tag{4.5}$$

ρ is the pheromone evaporation factor, $0 < \rho < 1$.

$$\Delta\tau_{ij}(t) = \ \sum_{k=1}^{m}\Delta\tau_{ij}^{k}(t) \tag{4.6}$$

4.3. General Pheromone Update. Once all the ants have completed an iteration, the pheromones of the bridges (i, j) corresponding to the preferable path are reinforced by a quantity Q/L_{best}, where L_{best} is the duration of the best path found T_{best}. This reinforcement is called general pheromone updating and is calculated using equation 4.7:

$$\tau_{ij}(t) = \ \tau_{ij}^{(t)} + \frac{Q}{L_{best}} \quad \text{si}(i,j) \in T_{best} \tag{4.7}$$

5. Experiments and Results. In a cloud computing environment, we opted to compare the basic ant colony optimization algorithm with a modified version for task scheduling and resource allocation using the Cloudsim simulator. The simulations were conducted with varying parameters including the number of cloudlets and VMs, utilizing 40 ants for the experiments.

5.1. Total response time compared to tasks. In the first simulation result, we estimated the average response time by varying the quantity of virtual machines (VM) and the effective of cloudlets. Figure 5.1 shows the simulation result for a number of cloudlets less than or equal to 100 with a number of virtual machines set to 10.

The result of the simulation for a number of cloudlets between 100 and 1000 on 50 virtual machines is given in figure 5.2.

For the simulations whose results are given in figure 5.3, we fixed the number of cloudlets to 500 and varied the number of virtual machines (between 20 and 50 respectively).

The first graphs (Figures 5.1, 5.2, et 5.3) concerning the total response time of queries show that the response time decreases remarkably with a high gain when using the modified ACO optimisation algorithm.

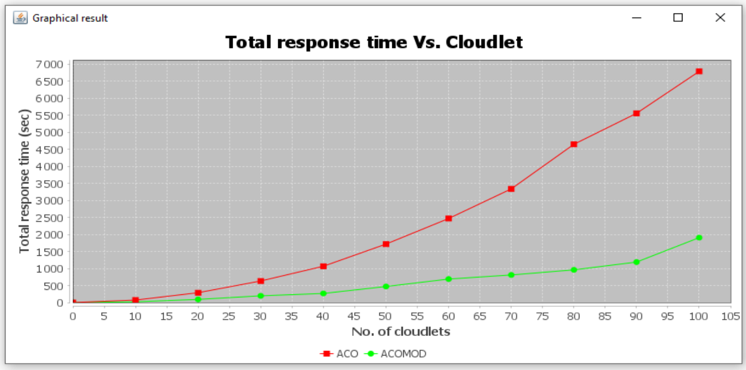

Fig. 5.1: The first total response time

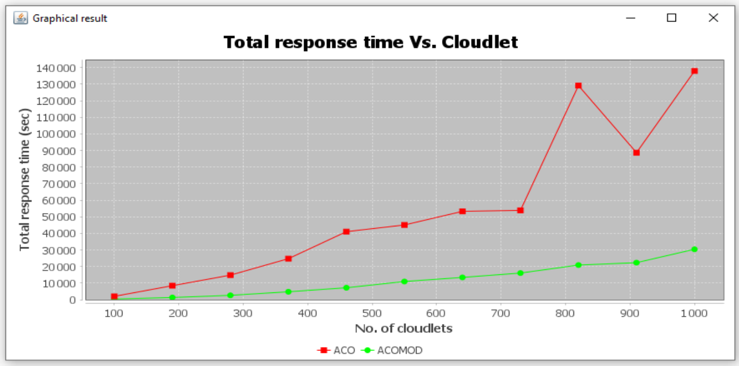

Fig. 5.2: The second total response time

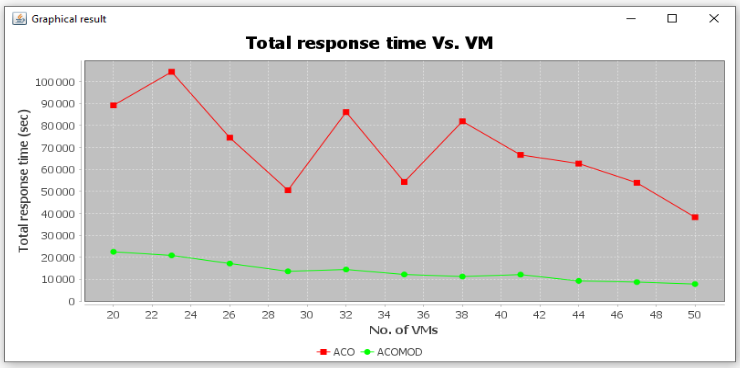

Fig. 5.3: The third total response time

5.2. Total execution cost. For this series of simulations, we calculated the total execution cost by varying the quantity of virtual machines (VM) and the effective of cloudlets. Figure 5.4 shows the simulation result for a number of cloudlets less than or equal to 100 with a number of virtual machines set to 10.

The result of the simulation for a number of cloudlets between 100 and 1000 on 50 virtual machines is

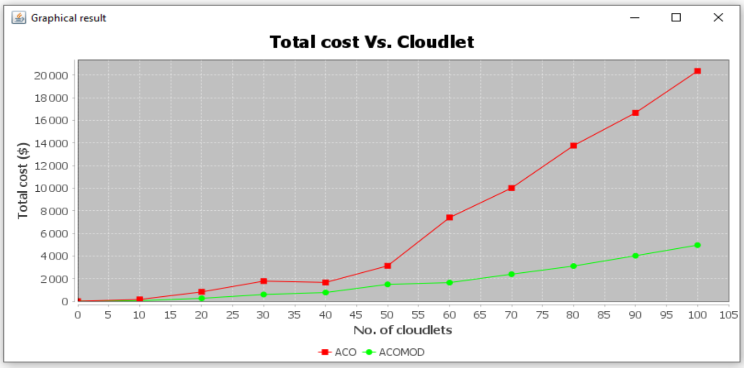

Fig. 5.4: The first total execution cost

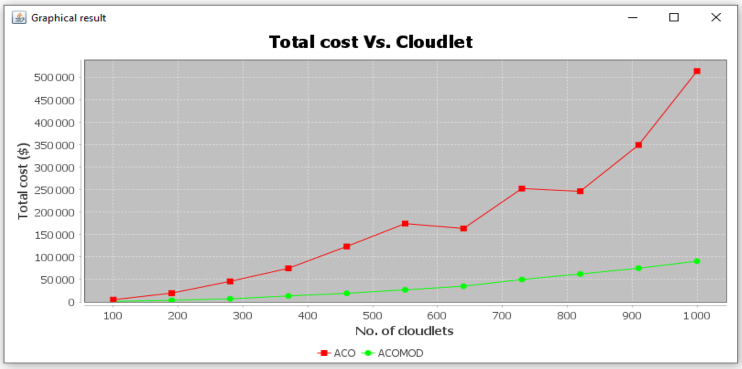

Fig. 5.5: The second total execution cost

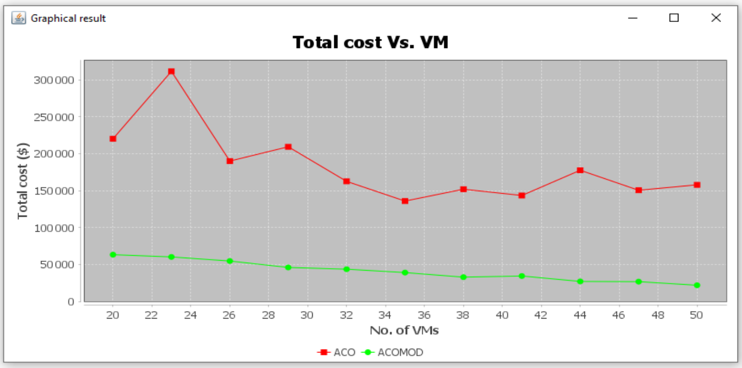

Fig. 5.6: The third total execution cost

given in figure 5.5. For the simulations whose results are given in Figure 5.6, we varied the number of virtual machine (VM) (between 20 and 50) and we fix the number of cloudlets to 500.

In the graphs in (Figures 5.4, 5.5 and 5.6, we can see that the cost of executing queries has been reduced considerably, with a significant gain.

5.3. Average response time. For this simulation series, we calculated the average response time by increasing the quantity of virtual machines (VM) and the effective of cloudlets. Figure 5.7 shows the simulation result for a number of cloudlets between 1000 and 2000 with a number of virtual machines set to 100.

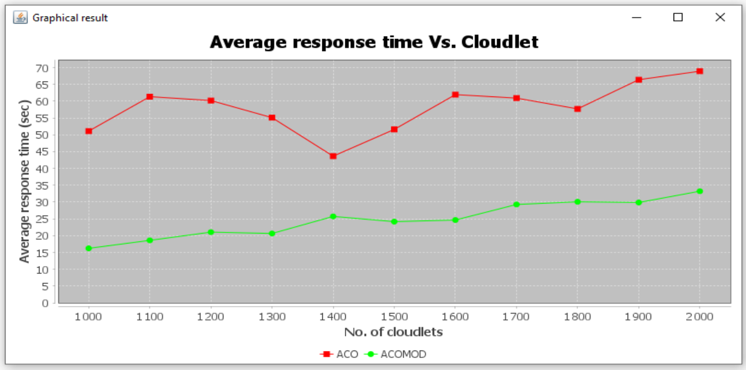

Fig. 5.7: The average response time

In the graphs in (Figures 5.7, we can see that the cost of executing queries has been reduced considerably, with a significant gain.

6. Conclusion. Cloud computing technology evoke an original model for the provision of computing resources. This technology has several advantages such as resource elasticity, pay-per-use payment model and others benefits.Cloud computing makes it possible to make resource allocations IT, but resources are not always sufficient. They often cannot meet user needs. Therefore, task scheduling and resource allocation mechanisms are needed to improve the optimization criteria. New strategies can employ some scheduling and allocation concepts to provide better task scheduling and resource allocation. A scheduling problem consists of organizing the execution of tasks over time, taking into account time constraints (deadlines, sequence constraints) and constraints relating to the availability of the required resources. The ACO algorithm, imitated from the act of foraging ants, which is a popular fuzzy optimization technique. The major principle of this act is the hidden dialogue between the ants by means of trails of chemical pheromones which allow them to find short routes between their home and their food germs. In this framework, the decisions made by all the ants are relevant and the experiences of all the ants are used at each iteration to build the new optimal solution. In our modest work, we have proposed a new modified variety of the ACO algorithm. We tested our algorithm under the CloudSim simulator. The approach reduces the total response time, the average response time, and the total execution cost. In summary, the simulation results obtained for the proposed scheduling strategy are satisfactory, very encouraging, and meet the objectives set. The improved ACO algorithm for task scheduling is not tested on real workflows. This point is very important to consider in future work. We also propose an improvement of the algorithm by hybridization with the gray wolf optimization algorithm which is an algorithm widely used recently in this type of work.

REFERENCES

[1] Ian Foster, Yong Zhao, Ioan Raicu, and Shiyong Lu, *Cloud Computing and Grid Computing 360-Degree Compared*, 2009.

[2] Marco Dorigo and Thomas Stützle, *The Ant Colony Optimization Metaheuristic: Algorithms, Applications, and Advances*, 2003.

[3] Liyun Zuo, Lei Shu, Shoubin Dong, Chunsheng Zhu and Takahirot Hara, *A Multi-Objective Optimization Scheduling Method Based on the Ant Colony Algorithm in Cloud Computing*, vol. 3, pp. 2687-2699, IEEE, December 2015.

[4] A. Arunarani and D.Manjula *Task scheduling techniques in cloud computing a literature survey*, Future Generation Computer Elsevier, vol. 91, pp. 407–415, 2019.

[5] S.K. Pande, S.K.Pande, and S. Das, *Task partitioning scheduling algorithms for heterogeneous multi-cloud environment*, Arab J. Sci. Eng. 43, pp. 913–933, 2018.

[6] R.P.Padhy, M.R.Patra and S.C.Satapathy, *Cloud computing: security issues and research challenges*, IRACST Int. J. Comput. Sci. Inf. Technol. Secur.,1(2), pp. 136–145, 2011.

[7] F. Yiqiu, X. Xia, and G. Junwei *Cloud computing task scheduling algorithm based on improved genetic algorithm*, IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC), Chengdu, China, pp. 852–856, 2019.

[8] A. Gupta and R. Garg *Load balancing based task scheduling with ACO in cloud computing*, International Conference on Computer and Applications (ICCA), Doha, pp. 174–179, 2017.

[9] M. Gawali and S. Shinde, *Task scheduling and resource allocation in cloud is using a heuristic approach*, J. Cloud Comput. Adv. Syst. Appl.,7(4), pp. 1–16, 2018.

[10] Arvind Lal and C. Rama Krishna, *Critical Path-Based Ant Colony Optimization for Scientific Workflow Scheduling in Cloud Computing Under Deadline Constraint*, Ambient Communications and Computer Systems, Springer Singapore, pp. 447-461, 2018.

[11] Archana N. Jethava and Megha R. Desai, *Optimizing Multi Objective Based Dynamic Workflow Using ACO and Black Hole Algorithm in Cloud Computing*, 3rd International Conference on Computing Methodologies and Communication (ICCMC), pp.1144-1147, 2019.

[12] M. Faisal and F. Albogamy, *Ant Colony Optimization Algorithm Enhancement for Better Performance*, IEEE World AI IoT Congress (AIIoT), pp. 0701-0710, 2023.

[13] C. Yin, Q. Fang, H. Li, Y. Peng,X. Xu and D. Tang, *An optimized resource scheduling algorithm based on GA and ACO algorithm in fog computing*, The Journal of Supercomputing, vol.80, pp. 4248–4285, 2024.

[14] A. Verma and S. Kaushal, *A hybrid multi-objective Particle Swarm Optimization for scientific workflow scheduling*, The Journal of Parallel Computing, vol.62, pp.1-19, 2017.

[15] L. Zimny, U. Schroeders, and O. Wilhelm, *Ant colony optimization for parallel test assembly*, The Journal of Behavior Research Methods, 2024.

[16] L. Bei, L. Wenlin, S. Xin and X. Xibin, *An improved ACO based service composition algorithm in multi-cloud networks*, The Journal of Cloud Computing, vol.13(17), 2024.

[17] H. Liu, A. Lee, W.Lee, P. Guo, *DAACO: adaptive dynamic quantity of ant ACO algorithm to solve the traveling salesman prob*, The Journal of Complex & Intelligent Systems, vol.9(4), 2023.

[18] A. J. Mohammed, K. I. Ghathwan and Y. Yusof, *Optimal robot path planning using enhanced particle swarm optimization algorithm*, Iraqi Journal of Science, pp. 178-184, 2020.

[19] C. Dudeja, *Fuzzy-based modified particle swarm optimization algorithm for shortest path problems*, Soft Computing, vol. 23, pp. 8321-8331, 2019.

[20] T. Chakraborty, S. Chakrabarti , and B. Hazra, *Study of β, ρ and Q_0 parameters for Shortest Path Estimation using Ant Colony Optimization*, 2019 IEEE Region 10 Symposium (TENSYMP), pp. 813-818, 2019.

[21] H. Alayed, F. Dahan, T. Alfakih, H. Mathkour, and M. Arafah, *Enhancement of Ant Colony Optimization for QoS-Aware Web Service Selection*, IEEE Access, vol. 7, pp. 97041–97051, 2019.

[22] J. Chen and J. Zhou, *An Improved Ant Colony Optimization for QoS-Aware Web Service Composition*, In: 2020 Eighth International Conference on Advanced Cloud and Big Data (CBD), pp 20–24, 2020.

[23] P. Shobeiri, M. Akbarian Rastaghi, S. Abrishami, and B. Shobiri. *PCP–ACO: a hybrid deadline-constrained workflow scheduling algorithm for cloud environment.* The Journal of Supercomputing, vol 80(6), pp. 7750-7780, 2024.

[24] R. Ghafari and N. Mansouri, *Improved Harris Hawks Optimizer with chaotic maps and opposition-based learning for task scheduling in cloud environment*, Cluster Computing, vol. 27(2), pp. 1421-1469, 2024.

[25] J. Kok Konjaang and Lina Xu, *Meta-heuristic Approaches for Effective Scheduling in Infrastructure as a Service Cloud: A Systematic Review*, Journal of Network and Systems Management, Vol. 29(2), 2021.

Edited by: Kavita Sharma
Special issue on: Recent Advance Secure Solutions for Network in Scalable Computing
Received: May 15, 2024
Accepted: Jul 23, 2024